CLAIMS:

What is claimed is:

1. A method of managing system reset interrupts, comprising:

receiving a system reset interrupt for a logical partition;

determining if an operation on the logical partition is being performed at the time the system reset interrupt is received; and

deferring handling of the system reset interrupt until after the operation on the logical partition is completed.

2. The method of claim 1, further comprising storing contents of registers of a processor associated with the logical partition in a context memory buffer in response to receiving the system reset interrupt.

3. The method of claim 1, further comprising determining if the system reset interrupt is a hard reset.

4. The method of claim 3, wherein if the system reset interrupt is a hard reset, a system reset interrupt delay flag is cleared and the processor is reallocated to a global processor available pool.

5. The method of claim 1, wherein the operation is a hypervisor call, and wherein deferring handling of the system reset interrupt until after the operation on the logical partition is completed includes:

setting a system reset interrupt flag; and

changing a return control instruction to a branch instruction that branches to a routine for simulating the system reset interrupt.

6. The method of claim 5, further comprising:

restoring register context for the processor after setting the system reset interrupt flag and changing the return control instruction; and

returning control to the operation.

7. The method of claim 1, wherein deferring handling of the system reset interrupt until after the operation on the logical partition is completed includes:

checking a system reset interrupt flag to determine if it is set; and

resetting a return control instruction to an original value in a jump table for a processor associated with the logical partition, if the system reset interrupt flag is set.

8. An apparatus for managing system reset interrupts, comprising:

means for receiving a system reset interrupt for a logical partition;

means for determining if an operation on the logical partition is being performed at the time the system reset interrupt is received; and

means for deferring handling of the system reset interrupt until after the operation on the logical partition is completed.

9. The apparatus of claim 8, further comprising means for storing contents of registers of a processor associated with the logical partition in a context memory buffer in response to receiving the system reset interrupt.

10. The apparatus of claim 8, further comprising determining if the system reset interrupt is a hard reset.

11. The apparatus of claim 10, wherein if the system reset interrupt is a hard reset, a system reset interrupt delay flag is cleared and the processor is reallocated to a global processor available pool.

12. The apparatus of claim 8, wherein the operation is a hypervisor call, and wherein the means for deferring handling of the system reset interrupt until after the operation on the logical partition is completed includes:

means for setting a system reset interrupt flag; and

means for changing a return control instruction to a branch instruction that branches to a routine for simulating the system reset interrupt.

13. The apparatus of claim 12, further comprising:

means for restoring register context for the processor after setting the system reset interrupt flag and changing the return control instruction; and

means for returning control to the operation.

14. The apparatus of claim 8, wherein the means for deferring handling of the system reset interrupt until

after the operation on the logical partition is completed includes:

means for checking a system reset interrupt flag to determine if it is set; and

means for resetting a return control instruction to an original value in a jump table for a processor associated with the logical partition, if the system reset interrupt flag is set.

15. A computer program product in a computer readable medium for managing system reset interrupts, comprising:

first instructions for receiving a system reset interrupt for a logical partition;

second instructions for determining if an operation on the logical partition is being performed at the time the system reset interrupt is received; and

third instructions for deferring handling of the system reset interrupt until after the operation on the logical partition is completed.

16. The computer program product of claim 15, further comprising fourth instructions for storing contents of registers of a processor associated with the logical partition in a context memory buffer in response to receiving the system reset interrupt.

17. The computer program product of claim 15, further comprising fifth instructions for determining if the system reset interrupt is a hard reset.

18. The computer program product of claim 17, wherein if the system reset interrupt is a hard reset, a system

reset interrupt delay flag is cleared and the processor is reallocated to a global processor available pool.

19. The computer program product of claim 15, wherein the operation is a hypervisor call, and wherein the third instructions for deferring handling of the system reset interrupt until after the operation on the logical partition is completed include:

instructions for setting a system reset interrupt flag; and

instructions for changing a return control instruction to a branch instruction that branches to a routine for simulating the system reset interrupt.

20. The computer program product of claim 19, further comprising:

instructions for restoring register context for the processor after setting the system reset interrupt flag and changing the return control instruction; and

instructions for returning control to the operation.

21. The computer program product of claim 15, wherein the third instructions for deferring handling of the system reset interrupt until after the operation on the logical partition is completed include:

instructions for checking a system reset interrupt flag to determine if it is set; and

instructions for resetting a return control instruction to an original value in a jump table for a processor associated with the logical partition, if the system reset interrupt flag is set.